Plan For the Week

- **Solve problems by programming in Python**
  - Like to do "real-world" problems, but we're very new to the language
  - Learn the syntax and semantics of simple Python programs
- **Compsci 101 Way-of-life**
  - Assignments, APTs, READMEs, oh-my
- **Vocabulary and Concepts**
  - Talk, think, speak like a Compsci 101 student
  - Talk, think, speak like a computer scientist?

PFTDay (introducing the week)

- **Review Python program we didn't get to**
  - helloaroundtheworld.py
  - Forming hypotheses about programs and Python, testing hypotheses by running code

- **Functions in Python and Compsci101**
  - What is a function, organizing programs
  - How do we gain understanding and practice with Python functions in Compsci101

Understanding terminology: code

- **Move from "Hello World" to "Hello Around the World"**
  - **Look at Python, code, libraries**
  - **Learning (reviewing) terminology about Python**

```
print "hello world"
```

```
f = open("hello_unicode.txt")
for line in f:
    print line)
```

Running and Understanding Code

- **Need Python compiler/interpreter**
  - We're using Canopy, includes libraries
- **Need an editor development environment**
  - We use Eclipse and PyDev, open source and widely used, Ambient is Duke Plugin
- **You need experience thinking and coding and debugging ideas and code:**
  - Installing the suite of tools can be cumbersome
    - Persist, Perservere, Get Help, start over ☹

Code Dissection

- **Every line thought about, occasionally understood at different levels**
  - **Use your understanding of natural language and experience, apply to Python**

```
f = open("hello_unicode.txt")
```

  - **Run program and apply knowledge to each of the other lines**

```
f = open("hello_unicode.txt")
for line in f:
    print line
```

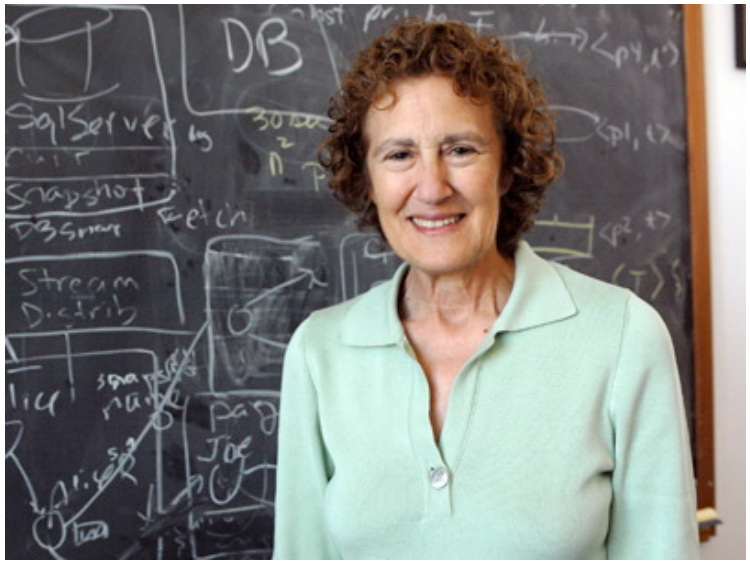
Questions about Python Code

- **Every line thought about, occasionally understood**
  - **What about when you make changes to the program? Answer these questions about possible errors introduced when making changes**

http://bit.ly/101fall15-827-3

Barbara Liskov

- (one of) first women to earn PhD from compsci dept
  - Stanford 1968
- Turing award in 2008
  - Programming Languages



"It's much better to go for the thing that's exciting. But the question of how you know what's worth working on and what's not separates someone who's going to be really good at research and someone who's not. There's no prescription. It comes from your own intuition and judgment."

Hello Around the World in Python

- We open a file, and we open a URL
  - Syntax slightly different, concept is similar
  - Real-world differences between files and URLs?

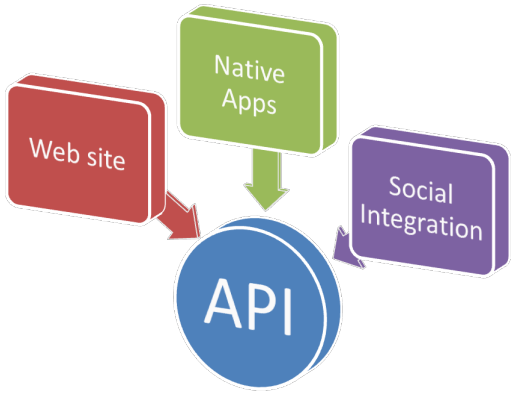
```
f = open("hello_unicode.txt")
```

```
f = urllib2.urlopen("http://nytimes.com")
```

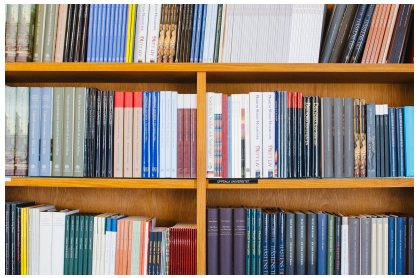
- Must adhere to syntactic rules of Python
  - Naming, whitespace, : or . or (or) or [or]
- Must adhere to semantic rules of Python
  - Can't loop over anything, more rules to follow

Libraries, Modules, APIs

- We need libraries of useful code
  - Can't write it all from scratch
  - Connect to web
  - Payment systems
  - Mapping systems
  - Geolocation ...



- Application Programming Interface
  - How to "connect" to software
  - Syntax and semantics of use

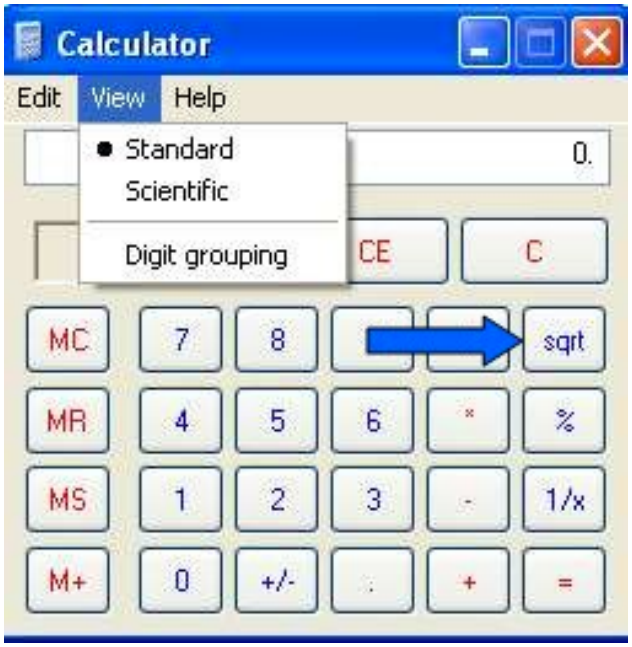


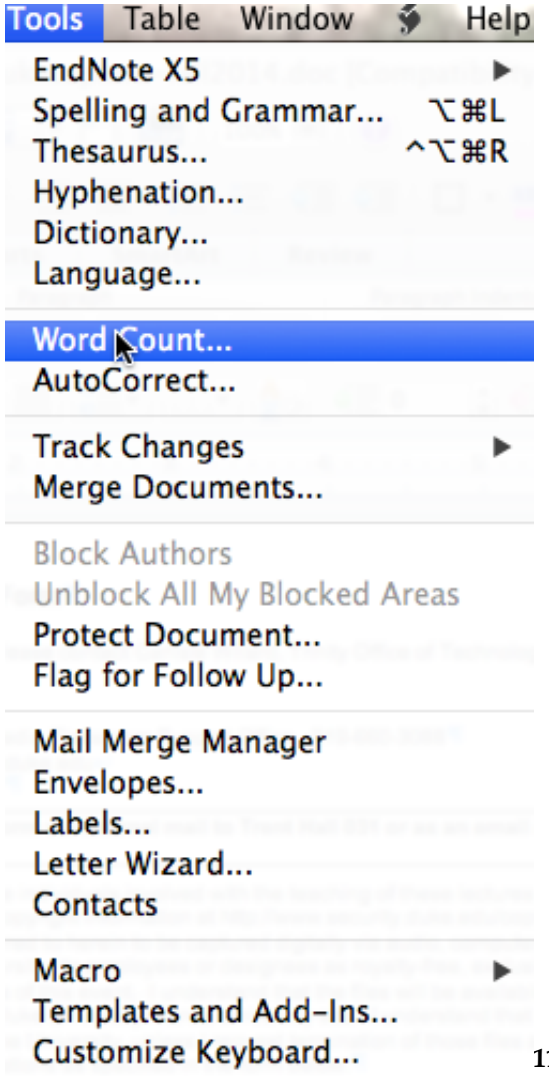
Python programming

- **You write a Python program**
  - Starting from scratch, snarfing, finding online
  - You execute it using a Python interpreter
  - Code executes top-to-bottom, see Hello World

- **Constructing a Python program**
  - Like constructing sentences in a new language
  - Learning and using vocabulary
  - Constructing increasingly complex programs, but starting from something simple

Examples of functions





Functions explained

- In a calculator, sqrt: number in -> number out
  - What is domain, what is range?
- In MSWord, word count: document -> number
  - Domain is word doc, range is integer
- In browser, web: URL -> HTML formatted "page"
  - Domain is valid URL, range is HTML resources
- In Python we see similar structure!

Python Functions

- **Answer these questions based on thinking, don't run any code**
  - http://bit.ly/101fall15-901-1

- **Why do we need functions?**
  - **Manage complexity of large programs**
  - **Test and develop code independently**
  - **Reuse code in new contexts: create APIs!**

Functions return values

- **Most functions return values**
  - Sometimes used to make things simpler, but returning values is a good idea

```
def inch2centi(inches):
    return 2.54*inches

xh = inch2centi(72)
```

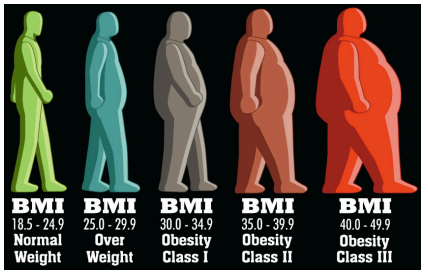
```
def pluralize(word):
    return word + "es"

pf = pluralize("fish")
```

What is an APT? BMI APT

- Automated/Algorithmic Problem Testing
  - Write one function, 2-30 lines, solve a problem
  - Tested automagically in the browser
  - Test test test … Quality of code not an issue

- Start simple, build toward more complex
  - What is a function? A function call?
  - What is a parameter? Argument?
  - How do you run/execute a program



How to solve an APT

- Two very, very, very important steps
  1. How to solve the problem without computer
     Paper, Pencil, (Calculator)
  2. How to translate problem-solving to Python

- Both steps can be hard, vocabulary and language are initially a real barrier
  - More Python experience, easier step 2 becomes
  - With experience, step 2 can influence step 1
- Step 1 is key, without it you won't get anywhere

Functions: BMI (Body Mass Index)

- What is formula? How to use it?
  - Functions allow us to re-use the formula
    - Make sure units are correct, formula right
  - What if we want to validate data?
  - What if we want to notify folks who might need guidance?

call replaced by return value, why use function?

```
def bmi(weight, height):
    return 703.07 * weight/(height*height)

if bmi(170,72) < 18.5:
   print "underweight"
```

What does return statement do?

- Programs execute one line at a time
  - After one statement finishes, the next executes
  - Calling a function causes its code to execute
    - What happens in the code that calls the function?

- The value returned *replaces* the function call
  - `print math.sqrt(25.0)`
  - `if bmi(170,72) < 18.5: print "underweight"`

- What if nothing returned?
  - `None` by default in Python

Toward creating functions

- **New meets old**
  - https://www.youtube.com/watch?v=0lM-NyN06rA

Old MacDonald had a farm, Ee-igh, Ee-igh, oh!
And on his farm he had a pig , Ee-igh, Ee-igh, oh!
With a oink oink here
And a oink oink there
Here a oink there a oink everywhere a oink oink
Old MacDonald had a farm, Ee-igh, Ee-igh, oh!

Creating Parameterized Function

What differs? Variable or Parameter

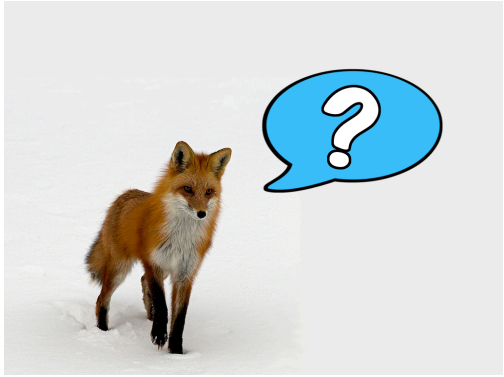
Old MacDonald had a farm, Ee-igh, Ee-igh, oh!
And on his farm he had a ***horse***, Ee-igh, Ee-igh, oh!
With a ***neigh neigh*** here
And a ***neigh*** neigh there
Here a ***neigh*** there a ***neigh*** everywhere a ***neigh neigh***
Old MacDonald had a farm, Ee-igh, Ee-igh, oh!

Old MacDonald had a farm, Ee-igh, Ee-igh, oh!
And on his farm he had a ***pig*** , Ee-igh, Ee-igh, oh!
With a ***oink oink*** here
And a ***oink oink*** there
Here a ***oink*** there a ***oink*** everywhere a ***oink oink***
Old MacDonald had a farm, Ee-igh, Ee-igh, oh!

Abstracting over code: functions

- http://goo.gl/DfcPgI
- See snarf for class work as well

- These functions do not return values, they print
  - Illustrates problem decomposition, but …
  - Normally have each function return a value
  - Normally use the return value in function call



Part of http://goo.gl/DfcPgl (and snarf)

```
def eieio():
  print "Ee-igh, Ee-igh, oh!"

def refrain():
  print "Old MacDonald had a farm,",
  eieio()

def had_a(animal):
  print "And on his farm he had a",animal,",",
  eieio()
```

Lots of commas

Anatomy and Dissection of Print

- Print generates output to a console, window, …
  - Depends on how program invoked
  - Basically used for: help with debugging and creating output for copy/paste, view

```
print "hello,",x,"what's up",y
```

- Space inserted between comma-separated items
  - Can use string concatentation, `"hello"+str(x)`
  - If statement ends with comma, no newline
  - Print anything that has a string representation…

Tracing program execution

- The `def` statement defines a function
  - Creates a name that can be used in program
  - Name encapsulates program statements, creates its own environment for running code
    - Variables, parameters, *local* to the function

- Function name and statements part of Python execution environment
  - Can *call* or *invoke* the function
  - If *parameters* needed, must *pass* values for each
  - Visualize program execution: PythonTutor, brain

Abstraction over barnyards

- In OldMacPrint we have `pig()` and `fox()` ...
  - What's the same in these? What's different?
  - Capture differences in parameters/variables
- Create new function:
  - `def verse(animal, noise)`

- Look at pig() and fox() create new function
  - Call: `verse("horse", "neigh")`
  - Call: `verse("cow", "moo")`

http://bit.ly/101fall15-901-2

Nathan Myhrvold

- **Who is he?**
  - http://bit.ly/myhrvold

We invent for fun. Invention is a lot of fun to do. And we also invent for profit. The two are related because the profit actually takes long enough that, if it isn't fun you wouldn't have the time to do it.

http://bit.ly/zwlqxn